CENTRE DE RECHERCHE
SUR LES TRANSPORTS
(C.R.T.)
CENTRE FOR RESEARCH
ON TRANSPORTATION

UNIVERSITÉ DE MONTRÉAL
ÉCOLE DES HAUTES ÉTUDES COMMERCIALES
ÉCOLE POLYTECHNIQUE

CENTRE DE RECHERCHE SUR LES TRANSPORTS
UNIVERSITÉ DE MONTRÉAL
C.P. 6128, SUCCURSALE A,
MONTRÉAL
CANADA H3C 3J7

TÉLÉPHONE: (514) 343-7575
TÉLEX: 05-24146 BIBPOLYTEC
EMAIL: crt@crt.umontreal.ca
TÉLÉCOPIEUR (FAX): (514) 343-7121

90-06643/

CRT-695

COMPUTATIONAL EXPERIMENTS WITH EXCHANGE PROCEDURES FOR THE MULTIPLE TRAVELING SALESMEN PROBLEM WITH TIME WINDOWS

by

Jean-Yves Potvin
Jean-Marc Rousseau

Centre de recherche sur les transports - Publication #695
May 1990

Dépôt légal - 2e trimestre 1990

Abstract. In this paper, we present and compare numerical results with various iterative improvement procedures for Multiple Traveling Salesmen Problems with Time Windows (MTSPTW). We concentrate on the few procedures that have been claimed to be particularly efficient for problems with time windows. Our experiments on a set of randomly generated euclidean problems show that a hybrid OR-opt and 2-opt* approach outperforms all other strategies, including the well known 3-opt exchange procedure, which has a higher order complexity. This hybrid exchange strategy is thus very powerful for improving initial feasible solutions when time windows are to be satisfied.

Key Words. Multiple Traveling Salesmen Problem, Exchange heuristics, Time windows.

Résumé. Dans cet article, nous présentons des résultats expérimentaux obtenus avec diverses stratégies d'amélioration itérative pour le problème des M-Voyageurs de Commerce avec fenêtres de temps. Nous nous intéressons aux quelques stratégies qui se sont révélées particulièrement efficaces pour de tels problèmes. Les résultats obtenus avec des problèmes euclidiens générés aléatoirement démontrent qu'une approche hybride OR-opt et 2-opt* produit de meilleures solutions que la stratégie classique d'échange 3-opt, qui est pourtant de complexité supérieure. Cette approche hybride est donc particulièrement efficace lorsqu'il s'agit d'améliorer des solutions initiales réalisables pour des problèmes avec fenêtres de temps.

Mots Clés. Problème des M-Voyageurs de Commerce, Heuristiques d'échange, Fenêtres de temps.

Section 1. Introduction

Routing and Scheduling problems are known to be a very important part of any distribution system. It is thus not surprising that the Vehicle Routing Problem (VRP) has received a lot of attention in the past. This problem is concerned with the design of routes for a fleet of vehicles. Each route starts and ends at a central depot and services a set of customers with known demands. The objective is to build routes that service all customers at minimum cost and that do not exceed vehicle capacity [2].

In the Vehicle Routing and Scheduling Problem with time windows (VRSPTW), these issues must be addressed with the additional complexity of allowable delivery times or time windows at each customer. With time windows, the routing cost does not only include total travel time, but also the waiting time incurred when a vehicle arrives too early at a customer location. Hence, temporal issues are an important concern when solving these problems. In this context, a solution is feasible if the total demand on each route does not exceed vehicle capacity and if each customer is serviced before his(her) time window's upper bound. An efficient route building procedure for this type of problem is described by Solomon in [7].

In this paper, we are mainly concerned with problems where the capacity constraint is replaced by a "scheduling horizon". More precisely, a time window is introduced at the depot location and each vehicle's tour must start and end at the depot within the bounds of that window. Such a scheduling horizon acts similarly as a capacity constraint (e.g. by enlarging the scheduling horizon, more customers can be serviced by the same vehicle). In the literature, this class of problem is referred to as the Multiple Traveling Salesmen Problem with Time Windows (MTSPTW). We chose to work with MTSPTW problems rather than VRSPTW problems, because demand distribution among customers can have an impact on the strategy to be used, and we wanted here to concentrate on time window issues.

The paper analyzes and compares various iterative route improvement procedures in the MTSPTW context. Only a few procedures have proven to be useful when time windows are to be considered, and our objective is to compare them on a set of problems with tight and large time windows.

The rest of the paper is organized as follows. Section 2 first introduces general considerations about iterative route improvement procedures. Section 3 then explains why classical K-opt exchange procedures are not well adapted to problems with time windows. We also explain how to modify the 2-opt procedure in order to get the so-called 2-opt* procedure, which has proven to be very good for such problems [6]. Section 4 briefly describes the OR-opt and OR-opt-1 exchange procedures [5]. Section 5 then suggests a hybrid OR-opt and 2-opt* approach for improving MTSPTW solutions. Finally, in section 6, we report computational results on a set of randomly generated problems with tight and large time windows.

Section 2. Iterative Improvement Procedures

There exist many different ways to modify a solution in order to get a new improved solution. Usually, the modification step is embedded within a generic iterative improvement procedure of the following type:

1) Generate an initial solution.
2) Modify the current solution so as to reduce its cost.
3) Repeat step 2 until no more improvement is possible.

One of the best known strategy of this type is the K-opt exchange procedure of Lin [4]. In step 2 of the generic framework, it exchanges k links in the current route for k new links. If the route covers N links, there are $O(N^k)$ ways to select the k links to be replaced and there are consequently $O(N^k)$ new solutions that can be considered for potential improvement. This set of new solutions is often called the "neighborhood" of the current solution.

Although the complexity of step 2 of the generic framework is typically polynomial (e.g. $O(N^k)$ for a K-opt exchange procedure), the number of times it is repeated until no improvement is found, is exponential in the worst case [3]. However, apart from some rare pathological cases, this methodology has proven to converge quickly towards a local optimum, and it is thus widely used in everyday practice. In the following, we will refer to an $O(N^k)$ exchange procedure, when step 2 of the generic framework is in the order of $O(N^k)$.

For constrained problems, any modification to the current solution must be checked for feasibility. In the worst case, the service time at each customer has to be reevaluated in order to assess feasibility for problems with time windows. The feasibility check is thus itself in the order of O(N), and any improvement procedure with a complexity in the order of $O(N^k)$ for unconstrained problems, is really working in the order of $O(N^{k+1})$ for MTSPTW problems.

In the following sections, we concentrate on the 2-opt* [6], OR-opt and OR-opt-1 [5] exchange procedures, which are known to be powerful methods for problems with time windows. Since their complexity is in the order of $O(N^3)$ for such problems, they can be used to get quality solutions in a reasonable amount of time. In this paper, the 3-opt exchange procedure is used for comparison purposes only. Its complexity is in the order of $O(N^4)$ for problems with time windows and is thus too computationally expensive for most practical purposes (i.e. large "real-life" problems with one hundred customers and more).

Section 3. The 2-opt* strategy

The classical K-opt exchange procedures are not well adapted to problems with time windows, because most exchanges do not preserve route orientation. Figure 1 shows an example for a classical 2-opt exchange procedure. Since customers are typically sequenced

according to their time window's upper bound (those with the lowest upper bounds being the first to be serviced), reversing some portion of a route will most likely produce a non feasible solution.

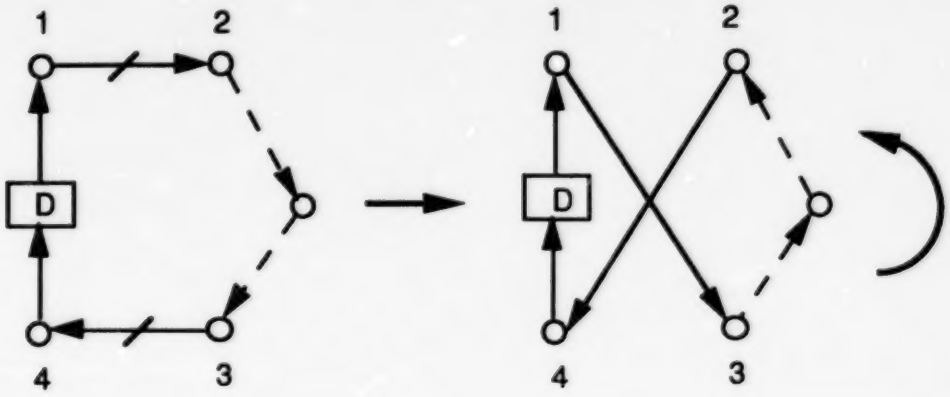

Figure 1. Exchange of links (1,2),(3,4) for links (1,3),(2,4)

In [6], we proposed a way to overcome this deficiency. Our approach is based on the well known fact that any Multiple Traveling Salesmen Problems (MTSP) can be converted into an equivalent Traveling Salesman Problem (TSP) [1,8]. We get that equivalence by creating M copies of the depot, each connected to the other nodes exactly as was the original depot. In such a network, any MTSP solution looks as a TSP solution (see figure 2a). We can thus consider a special 2-opt* exchange strategy that partitions the single tour into two distinct subtours (see figure 2b). Such a solution is feasible if and only if there is at least one copy of the depot in each subtour. This exchange strategy is particularly interesting for problems with time windows because it preserves route orientation and links the first customers on a route to the last customers of another route (and vice-versa). Such a solution is thus likely to be feasible.

The subtours can then be easily merged back into an equivalent single tour by exchanging two links connecting two customers to distinct copies of the depot (see figure 2c). If we look at the tour before and after the 2-opt* exchange, four different links have been replaced, including the exchange of two links for merging the two subtours into an equivalent single tour. The set of exchanges considered by the 2-opt* procedure is thus a subset of all exchanges considered by a classical 4-opt procedure.

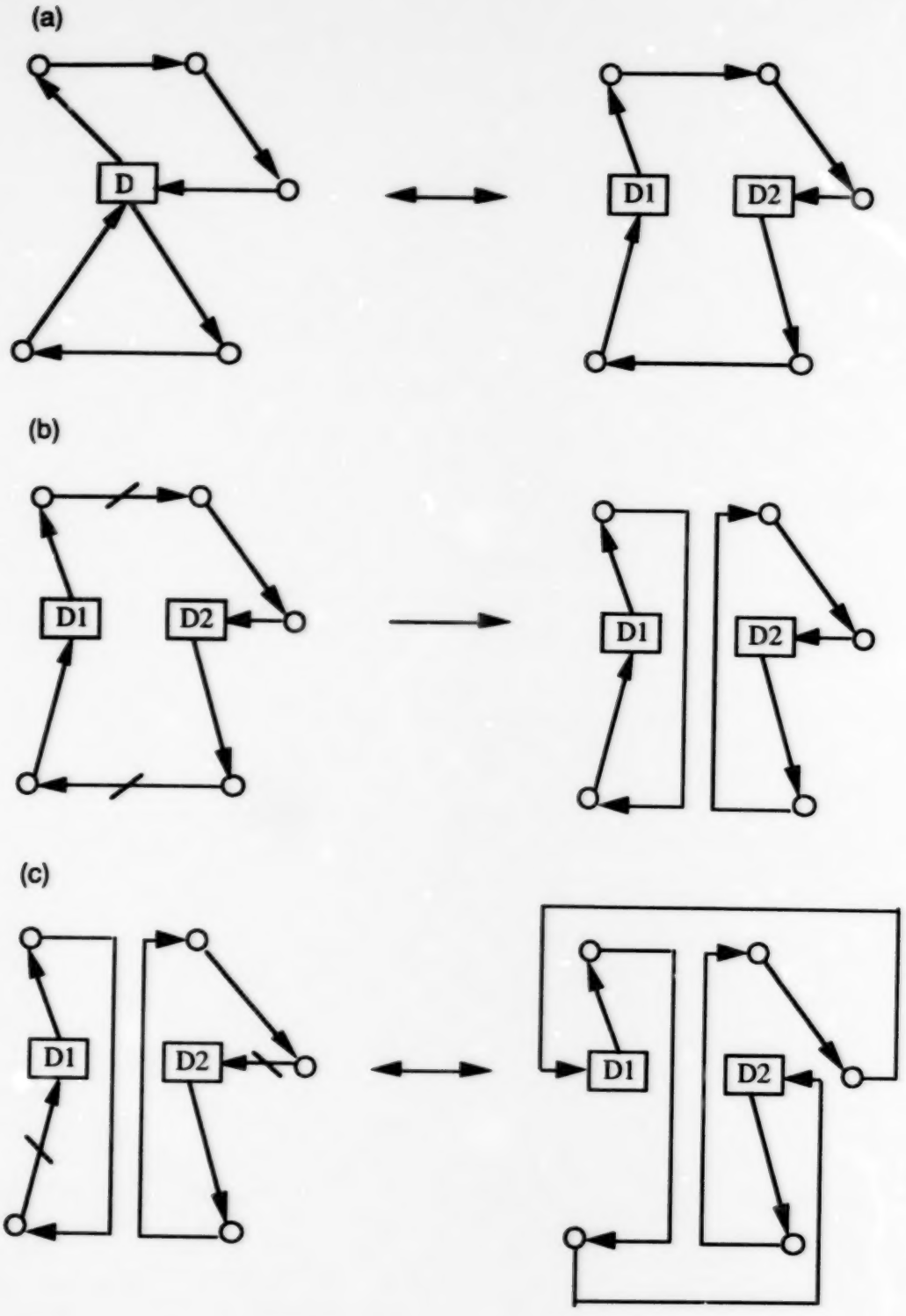

Figure 2. A 2-opt* exchange

We must observe that a 2-opt* strategy is suitable only for inter-route exchanges. It cannot be used for modifying the sequence of customers within a route, since any 2-opt* exchange involving two links in the same route would obviously generate one subtour without any copy of the depot. We can thus deduce that the 2-opt* strategy is not useful if the initial tour building algorithm produces an optimum clustering of the customers (i.e. if only sequencing adjustments within routes are needed for improving the solution). However, for the time being, there is no such heuristic algorithm for the MTSPTW, and inter-route exchanges are likely to produce substantial improvements over the initial solution.

In order to get inter-route as well as intra-route improvements, it is possible to apply a 2-opt exchange when the two links to be replaced are in the same route, and to apply a 2-opt* exchange otherwise. When time windows are not very tight, reversing some portion of a route may well be feasible and, in such a case, intra-route 2-opt exchanges may be valuable. In the following, we will refer to this exchange strategy as the mixed 2-opt/2-opt*.

The next section now presents two other powerful exchange strategies for problems with time windows, namely the OR-opt-1 and OR-opt exchange procedures.

Section 4. The OR-opt-1 and OR-opt strategies

These two route improvement procedures are well-known "node exchange procedures". An OR-opt-1 exchange procedure considers in turn each customer in the current solution and tries to reduce total cost by considering all other possible locations for that customer (see figure 3). The OR-opt exchange procedure generalizes the OR-opt-1 approach by considering in turn sequences of one, two and three adjacent customers in the current solution and tries to reduce total cost by considering all other possible locations for each sequence. Both strategies work in the order of $O(N^3)$ for problems with time windows, although the set of exchanges considered by the OR-opt-1 procedure is obviously a subset of those considered by the OR-opt procedure.

Although such strategies concentrate on nodes rather than links, they are in fact considering a subset of 3-opt exchanges. As mentioned before, most 3-opt exchanges do not preserve route orientation and

are likely to generate non feasible solutions. The OR-opt procedure, however, moves only small sequences of adjacent customers, which are typically "close" together from a spatial and temporal point of view, and inserts them at a new location while preserving route orientation. The OR-opt procedure thus focuses on a small subset of 3-opt exchanges that are more likely to be feasible. By this mean, the OR-opt procedure can generate solutions that almost compare with a 3-opt procedure for problems with time windows.

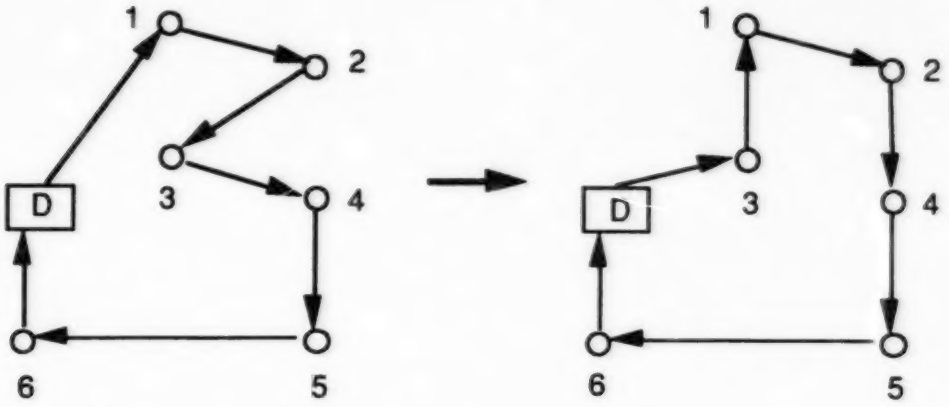

Figure 3. OR-opt-1 on customer 3 between depot D and customer 1

Both the OR-opt-1 and OR-opt exchange strategies are suitable either for inter-route or intra-route exchanges. However, they are particularly well suited for small adjustments within routes, because they introduce only slight perturbations into the current solution (i.e. sequences of only one, two or three adjacent customers are moved from one place to another).

In the following section, we now explain how to use the strenghts of the OR-opt and 2-opt* procedures for building a powerful hybrid exchange strategy.

Section 5. A hybrid OR-opt/2-opt* strategy

Some interesting points emerge from the previous description of the OR-opt and 2-opt* exchange procedures. The most important one is relating to their respective neighborhood. The OR-opt neighborhood is a subset of the 3-opt neighborhood, while the 2-opt* neighborhood is a

subset of the 4-opt neighborhood. Both methods, which are particularly powerful for problems with time windows, are thus exploring sets of exchanges that are "almost" disjoint (in some rare cases, a 2-opt* exchange can be equivalent to an OR-opt exchange).

In addition, a 2-opt* exchange introduces important inter-route modifications, while an OR-opt exchange introduces only small perturbations into the current solution. In other words, the 2-opt* exchange procedure moves into the solution space by considering new solutions that are very different from the current one, while the OR-opt focuses mainly on finer refinements. These considerations underline what could be gained by merging these two "complementary" approaches into a hybrid one.

In the current implementation, the hybrid OR-opt/2-opt* is based on two different improvement strategies. First, the OR-opt exchange procedure is applied so as to obtain an OR-opt optimal solution. The 2-opt* exchange procedure is then applied to the OR-opt optimal solution, so as to obtain a 2-opt* optimal solution. This is repeated until the solution stabilizes (i.e. when it is OR-opt *and* 2-opt* optimal). The second improvement strategy simply reverses the process and apply the 2-opt* exchange procedure before the OR-opt exchange procedure. The final result is then the best solution of the two.

The next section will now show the capabilities of this approach as compared with other approaches presented in this paper on a set of randomly generated problems.

Section 6. Computational Results

In order to assess the effectiveness of the new hybrid strategy, we performed computational experiments on a set of randomly generated MTSPTW problems with tight and large time windows. We ran C programs on an IBM-PC microcomputer so as to compare the following exchange strategies: OR-opt-1, OR-opt, 2-opt*, mixed 2-opt/2-opt*, mixed OR-opt/2-opt* and 3-opt. All exchange strategies were applied to the single "super-route", which is obtained by transforming a MTSP solution into an equivalent TSP solution (cf. section 3). The objective was to minimize total route time.

We created euclidean problems with N=25 and N=50 customers, by randomly generating their coordinates inside an area of 100 square kilometers, using a uniform distribution. The scheduling horizon was

set so as to obtain initial solutions with a small number of customers in each route for N=25 and a large number of customers in each route for N=50. Such initial solutions were created using the sequential algorithm of Solomon. The route initialization phase of Solomon's algorithm was done by selecting the farthest unrouted customer from the depot. We then generated four alternative solutions with four different sets of values for the parameters $(\alpha_1,\alpha_2,\mu,\lambda)$, as suggested by Solomon in his paper. The best overall solution was then selected as the initial solution for our improvement procedures (for a description of the sequential insertion procedure of Solomon and a definition of the above parameters, see [7]).

The center of each time window was randomly generated within the horizon, but feasibility checks were made to insure that vehicles could make a round-trip from the depot to any customer and satisfy the temporal restrictions. Large time windows were set so as to cover 20 to 25% of the scheduling horizon. Tight time windows were set so as to cover 5 to 10% of the horizon. As mentioned before, a waiting time is incurred when a vehicle arrives too early at a customer location, but a vehicle must never be too late.

The final results are presented in tables I and II. For each method, the average number of routes, total route time (in minutes) and computation time (in seconds) are shown. The relative improvement in total route time over the set of initial solutions is also shown, i.e.:

(initial solution - solution after exchange) / initial solution.

In each category, the average is taken over 40 problems for N=25 and over 20 problems for N=50.

Before discussing the results, we must first observe that the code has not been really optimized for any exchange strategy, so that better computation times are certainly achievable. Also, we did not want to use a screening test that would have greatly reduced computation time, but at the expense of overlooking some valid exchanges. The basic idea for such a screening test is the following: if we assume, for example, that a 2-opt exchange replaces links (k,l),(m,n) by the new links (k,m),(l,n) and that $t_{i,j}$ is the travel time from customer i to customer j, the test:

$$(t_{k,m} + t_{l,n}) - (t_{k,l} + t_{m,n}) < 0 \ ?$$

is certainly appropriate for screening good exchanges in the unconstrained case, since total route time cannot be reduced otherwise. The potential improvement of a given exchange can thus be evaluated before modifying the solution as such (i.e. before reevaluating service time at each customer). Unfortunately, for problems with time windows, a reduction in travel time can be associated with an even greater increase in waiting time, so that the overall route time is increased. Conversely, an increase in travel time can be associated with an even greater reduction in waiting time, so that the overall route time is reduced. We thus have to evaluate the new service time at each customer for each potential exchange in order to determine the total route time of the new solution, and this is computationally expensive. In practical settings, however, an exchange could be rejected when the increase in travel time is over some threshold, since it is then unlikely that total route time could decrease. Another alternative would be to minimize total travel time instead of total route time. However, this objective is rather "artificial" for problems with time windows, because solutions with low travel time and high waiting time are usually not acceptable.

From the results presented in Tables I and II, we can make the following remarks:

1) The best overall method with respect to solution quality is the hybrid OR-opt/2-opt* exchange strategy. Although more computationally expensive than the other $O(N^3)$ approaches, it generates slightly better solutions than a 3-opt, which is an $O(N^4)$ approach for problems with time windows. This result is very important, since it suggests that the brute force of a 3-opt strategy is not useful here for generating high quality solutions. The hybrid exchange strategy is much less computationally expensive and generates solutions of even better quality. Hence, this hybrid approach is really focusing on exchanges that are potentially useful for problems with time windows and should be used to generate high quality solutions in a reasonable amount of time.

2) There is no clear winner between OR-opt and mixed 2-opt/2-opt*, the two previously best known $O(N^3)$ exchange strategies for problems with time windows. 2-opt/2-opt* was slighty better with N=25 (a small number of customers in each route), but the

situation is reversed with N=50 (a large number of customers in each route).

3) The 2-opt* exchange strategy is very fast and is quite good for problems with tight time windows. It is thus an interesting alternative to consider for such problems, if a good solution has to be produced very quickly. However, its performance degrades for problems with larger time windows. By comparing with 2-opt/2-opt*, for example, such a degradation can be explained by the fact that reversing some portion of a route with a 2-opt is more likely to be feasible with larger time windows. Those opportunities are missed by 2-opt*.

4) Solutions produced by OR-opt-1 are very close to those produced by OR-opt. It suggests that if time windows are to be considered, there is little to be gained by extending the OR-opt-1 approach to sequences of two and three customers. Both OR-opt-1 and OR-opt take more computation time than 2-opt and mixed 2-opt/2-opt*, because the average number of iterations needed to reach a local optimum is higher. This could be explained by the fact that OR-opt-1 and OR-opt introduce only small perturbations into the solutions and thus progress more slowly towards a local optimum.

As a final remark, we should mention that some exchange strategies reduced the total number of routes for problems with a few customers in each route (cf. results with N=25). Although Solomon's route building algorithm was applied here with only one initialization criterion - in his paper, Solomon uses two different initialization criteria for generating alternative routes and takes the best overall solution - our results indicate that there is room for improving this algorithm. It would thus be an interesting research avenue to try to identify better route building heuristics for problems with time windows.

N=25 *time windows tight*	Number of routes	Route time (min.)	Route time Improvement	Computation time (sec.)
Initial Solution	4.45	1201.5	- - -	- - -
2-opt*	4.375	1131.0	5.9%	11.3
2-opt/2-opt*	4.375	1128.3	6.1%	23.8
OR-opt-1	4.425	1158.8	3.6%	22.4
OR-opt	4.375	1141.1	5.0%	53.3
OR-opt/2-opt*	4.325	1115.8	7.1%	141.8
3-opt	4.35	1117.1	7.0%	454.8

N=25 *time windows large*	Number of routes	Route time (min.)	Route time Improvement	Computation time (sec.)
Initial Solution	3.10	880.0	- - -	- - -
2-opt*	3.10	836.0	5.0%	7.2
2-opt/2-opt*	3.10	804.9	8.5%	16.6
OR-opt-1	3.10	815.8	7.3%	36.1
OR-opt	3.10	810.3	7.9%	59.5
OR-opt/2-opt*	3.05	762.4	13.4%	188.1
3-opt	3.075	791.7	10.0%	807.5

Table I Computational Results with N=25

N=50 *time windows tight*	Number of routes	Route time (min.)	Route time Improvement	Computation time (sec.)
Initial Solution	3.45	2066.0	- - -	- - -
2-opt*	3.45	2008.8	2.8%	59.7
2-opt/2-opt*	3.45	1978.8	4.2%	159.5
OR-opt-1	3.45	1979.7	4.2%	367.1
OR-opt	3.45	1976.7	4.3%	510.3
OR-opt/2-opt*	3.45	1944.1	5.9%	1273.2
3-opt	3.45	1960.4	5.1%	11656.2

N=50 *time windows large*	Number of routes	Route time (min.)	Route time Improvement	Computation time (sec.)
Initial Solution	2.50	1374.8	- - -	- - -
2-opt*	2.50	1351.7	1.7%	40.5
2-opt/2-opt*	2.50	1304.0	5.2%	94.4
OR-opt-1	2.50	1276.5	7.2%	381.7
OR-opt	2.50	1268.8	7.7%	547.4
OR-opt/2-opt*	2.50	1246.3	9.3%	1571.5
3-opt	2.50	1248.5	9.2%	19942.7

Table II Computational Results with N=50

Section 7. Conclusion

In this paper, we have described a powerful hybrid exchange strategy for MTSPTW problems. Computational results show that this approach outperforms the well known 3-opt exchange strategy, even if it has a lower order complexity. This paper has thus demonstrated that there is no need to use the brute force of the 3-opt exchange strategy for generating high quality solutions when time windows are the main concern.

Acknowledgements. We would like to thank Lucie Bibeau and Patrice Bergeron who ran and analyzed the experiments. Thanks also to Guy Lapalme for his useful comments. Financial support for this research has been provided by the Natural Sciences and Engineering Council of Canada (NSERC).

References

[1] Bellmore M. and Hong S. (1974), "Transformation of Multi-Salesman Problem to the Standard Traveling Salesman Problem", *J. ACM 21*, 500-504.

[2] Bodin L., Golden B., Assad A. and Ball M. (1983), "Routing and Scheduling of Vehicles and Crews: the State of the Art", *Computers and Operations Research 10(2)*.

[3] Johnson D.S., Papadimitriou C.H., Yannakakis M. (1985), "How easy is local search?", *IEEE Annual Symposium on Foundations of Computer Science*, 39-42.

[4] Lin S. (1965), "Computer Solutions of the Traveling Salesman Problem", *Bell System Tech. J. 44*, 2245-2269.

[5] Or I. (1976), "Traveling Salesman-type Combinatorial Problems and their Relation to the Logistics of Blood Banking", Ph.D. Thesis, Dept. of Industrial Engineering and Management Sciences, Northwestern University.

[6] Potvin J.Y., Lapalme G. and Rousseau J.M. (1989), "A Generalized K-opt Exchange Procedure for the MTSP", *INFOR 27(4)*, 474-481.

[7] Solomon M.M. (1987), "Algorithms for the Vehicle Routing and Scheduling Problem with Time Window Constraints", *Operations Research 35(2)*, 254-265.

[8] Svestka J. and Huckfeldt V. (1973), 'Computational Experience with an M-Salesmen Traveling Salesman Algorithm", *Management Science 19(7)*, 790-799.